

# Dynatrace - Quick Start Guide

¿Nuevo en el proyecto? Sigue estos pasos para configurar Dynatrace.

 **Tiempo estimado: 30 minutos**

## Pre-requisitos

- Acceso al tenant de Dynatrace de la organización
- Credenciales (PaaS Token, API Token)
- Docker instalado (recomendado)
- Backend y Frontend funcionando localmente

## **Paso 1: Obtener Credenciales (5 min)**

**Solicitar al equipo DevOps:**

### 1. Environment ID

Ejemplo: abc12345

### 2. Tenant URL

Ejemplo: <https://abc12345.live.dynatrace.com>

### 3. PaaS Token (para instalar OneAgent)

- Ir a: Settings > Integration > Platform as a Service
- Click: Generate token
- Permisos: InstallerDownload

Ejemplo: dt0c01.ST2EY72KQINXXXXXXXXXXXX...

### 4. API Token (para consultar alertas)

- Ir a: Settings > Integration > Dynatrace API
- Click: Generate token
- Permisos necesarios:
  - Read metrics (v2)
  - Read problems (v2)

- Write events (v2)
- Read entities (v2)

```
Ejemplo: dt0c01.XA7LQ9...XXXXXXXXXXXXXX...
```

## Paso 2: Backend - Configurar OneAgent (10 min)

### Opción A: Docker (Recomendado para desarrollo)

#### 1. Crear archivo de variables de entorno

```
cd svc-signature-router

# Crear archivo .env.dynatrace
cat > .env.dynatrace << EOF
DYNATRACE_ENV_ID=abc12345
DYNATRACE_URL=https://abc12345.live.dynatrace.com
DYNATRACE_PAAS_TOKEN=dt0c01.ST2EY72KQIN...
DYNATRACE_API_TOKEN=dt0c01.XA7LQ9...
DT_TAGS=environment=dev,application=signature-router,team=backend
ADMIN_PORTAL_ALERTS_MOCK=false
EOF
```

#### 2. Cargar variables

```
# En PowerShell
Get-Content .env.dynatrace | ForEach-Object {
    if ($_.Match '^([=]+)=(.*)$') {
        [Environment]::SetEnvironmentVariable($matches[1], $matches[2], 'Process')
    }
}

# En Bash/WSL
source .env.dynatrace
```

### 3. Actualizar application.yml

```
# svc-signature-router/src/main/resources/application.yml

dynatrace:
  url: ${DYNATRACE_URL:https://abc12345.live.dynatrace.com}
  api-token: ${DYNATRACE_API_TOKEN}

admin:
  portal:
    alerts:
      mock: false # ← Usar Dynatrace en lugar de mock
```

### 4. Instalar OneAgent localmente (Windows)

#### Opción 1: Via UI

1. Ir a Dynatrace: Deploy Dynatrace > Start installation > Windows
2. Descargar el instalador
3. Ejecutar:

```
.\Dynatrace-OneAgent-Windows.exe APP_LOG_CONTENT_ACCESS=1 INFRA_ONLY=0 /quiet
```

#### Opción 2: Via PowerShell

```
$env_id = $env:DYNATRACE_ENV_ID
$paas_token = $env:DYNATRACE_PAAS_TOKEN

Invoke-WebRequest `

-Uri
"https://$env_id.live.dynatrace.com/api/v1/deployment/installer/agent/windows/default/latest?Api-Token=$paas_token" `

-OutFile "Dynatrace-OneAgent-Windows.exe"

.\Dynatrace-OneAgent-Windows.exe APP_LOG_CONTENT_ACCESS=1 INFRA_ONLY=0 /quiet
```

### 5. Reiniciar aplicación

```
mvn spring-boot:run
```

#### Verificación Backend:

1. Ir a Dynatrace UI
2. Menú: Hosts
3. Buscar tu hostname

4. Verificar que aparece signature-router en Process Groups

5. Ir a: Services → Deberías ver signature-router-api

## 🌐 Paso 3: Frontend - Configurar RUM (10 min)

### 1. Registrar aplicación en Dynatrace

1. Ir a Dynatrace: Frontend

2. Click: Add new web application

3. Nombre: Signature Router Admin Panel

4. Application type: Single Page Application

5. Click: Create

#### Copiar los valores:

- Application ID: APPLICATION-1234567890ABCDEF
- JavaScript snippet (lo usaremos después)

### 2. Configurar variables de entorno

```
cd app-signature-router-admin

# Editar .env.local (o crear si no existe)
# Agregar:
NEXT_PUBLIC_DYNATRACE_ENV_ID=abc12345
NEXT_PUBLIC_DYNATRACE_APP_ID=APPLICATION-1234567890ABCDEF
NEXT_PUBLIC_DYNATRACE_APP_NAME=signature-router-admin
NEXT_PUBLIC_DYNATRACE_ENVIRONMENT=dev
NEXT_PUBLIC_USE_MOCK_DATA=false
```

### 3. Actualizar layout.tsx

```
// app-signature-router-admin/app/layout.tsx

import Script from 'next/script';

export default function RootLayout({ children }: { children: React.ReactNode }) {
  const dynatraceEnvId = process.env.NEXT_PUBLIC_DYNATRACE_ENV_ID;
  const dynatraceAppId = process.env.NEXT_PUBLIC_DYNATRACE_APP_ID;

  return (
    <html lang="es">
      <head>
```

```

{dynatraceEnvId && dynatraceAppId && (
    <Script
        id="dynatrace-rum"
        strategy="beforeInteractive"
        src={`https://js-
cdn.dynatrace.com/jstag/${dynatraceEnvId}/${dynatraceAppId}/ruxitagent.js`}
        crossOrigin="anonymous"
    />
) }
</head>
<body>{children}</body>
</html>
);
}

```

## 4. Reiniciar frontend

```
npm run dev
```

### Verificación Frontend:

1. Abrir navegador: `http://localhost:3001`
2. Abrir DevTools → Console
3. Ejecutar: `window.dtrum`
4. Debería retornar: `{initialized: true, ...}`
5. Ir a Dynatrace UI → Frontend → Tu aplicación
6. Deberías ver sesiones activas

## 💡 Paso 4: Verificar Integración de Alertas (5 min)

### 1. Implementar servicio de Dynatrace

**⚠ IMPORTANTE:** El archivo `AlertManagerServiceDynatraceImpl.java` debe ser creado (ver documentación completa en `INTEGRACION-DYNATRACE.md`)

### 2. Verificar configuración

```
# application.yml
admin:
  portal:
    alerts:
      mock: false  # ← DEBE estar en false
```

### 3. Probar en el frontend

```
# Abrir panel de alertas
http://localhost:3001/admin/alerts

# Debería mostrar:
# - Problemas abiertos de Dynatrace
# - NO las 5 alertas mock hardcodeadas
```

## Checklist Final

### Backend

- OneAgent instalado y running
- application.yml configurado con Dynatrace URL y API Token
- ADMIN\_PORTAL\_ALERTS\_MOCK=false en variables de entorno
- Backend arrancado y visible en Dynatrace UI (Hosts/Services)
- Logs muestran: [OneAgent] successfully connected

### Frontend

- Variables de entorno configuradas (.env.local)
- Script de RUM agregado en layout.tsx
- Frontend arrancado
- window.dtrum funciona en browser console
- Sesiones visibles en Dynatrace UI

### Alertas

- AlertManagerServiceDynatraceImpl implementado
- Mock deshabilitado
- Panel de alertas muestra problemas de Dynatrace
- Botones "Reconocer" y "Resolver" funcionan

# Troubleshooting

## OneAgent no aparece en Dynatrace

**Síntoma:** Backend está corriendo pero no aparece en Dynatrace UI

**Solución:**

1. Verificar que OneAgent está instalado:

```
# Windows  
Get-Service Dynatrace*  
  
# Debería mostrar: Dynatrace OneAgent (Running)
```

2. Verificar logs de OneAgent:

```
C:\ProgramData\dynatrace\oneagent\log\
```

3. Reiniciar OneAgent:

```
Restart-Service "Dynatrace OneAgent"
```

## RUM no funciona

**Síntoma:** window.dtrum es undefined

**Solución:**

1. Verificar que las variables de entorno están cargadas:

```
console.log(process.env.NEXT_PUBLIC_DYNATRACE_ENV_ID);  
// Debe mostrar: abc12345 (no undefined)
```

2. Verificar que el script se está cargando:

- Abrir DevTools → Network
- Filtrar por ruxitagent.js
- Debería aparecer con status 200

3. Verificar URL del script:

```
Debe ser:  
https://js-cdn.dynatrace.com/jstag/{env-id}/{app-id}/ruxitagent.js  
  
NO debe ser:  
https://js-cdn.dynatrace.com/jstag/undefined/...
```

## Panel de alertas muestra datos mock

**Síntoma:** Siempre muestra las mismas 5 alertas (HighErrorRate, ProviderDown, etc.)

**Solución:**

1. Verificar que mock está deshabilitado:

```
admin.portal.alerts.mock: false
```

2. Verificar que AlertManagerServiceDynatraceImpl existe y está activado:

```
@ConditionalOnProperty(name = "admin.portal.alerts.mock", havingValue = "false")
```

3. Verificar logs del backend:

```
Debe mostrar:
```

```
[DYNATRACE] Fetching problems from Dynatrace API
```

```
NO debe mostrar:
```

```
[MOCK] Using MOCK AlertManager Service
```

## Error 401 al consultar Dynatrace API

**Síntoma:** Logs muestran 401 Unauthorized al llamar API de Dynatrace

**Solución:**

1. Verificar que API Token tiene permisos correctos:

- Read problems (v2)
- Read metrics (v2)

2. Verificar formato del header:

```
// CORRECTO:  
headers.set("Authorization", "Api-Token " + apiToken);  
  
// INCORRECTO:  
headers.set("Authorization", "Bearer " + apiToken); // ✗ Wrong!
```

## Próximos Pasos

Una vez que la integración básica funcione:

1. **Dashboards**: Crear dashboards personalizados en Dynatrace
2. **Alerting**: Configurar alerting profiles y management zones
3. **SLOs**: Definir Service Level Objectives
4. **Synthetic Monitoring**: Crear health checks sintéticos
5. **Business Events**: Trackear eventos de negocio (signature created, payment completed)

Ver documentación completa en: [docs/INTEGRACION-DYNATRACE.md](#)

---

## Soporte

### ¿Problemas durante la configuración?

1. Revisar la documentación completa: [docs/INTEGRACION-DYNATRACE.md](#)
  2. Contactar al equipo DevOps
  3. Revisar logs de Dynatrace OneAgent
  4. Verificar la [documentación oficial de Dynatrace](#)
- 

¡Listo! 🎉

Ahora tienes Dynatrace funcionando con full-stack observability automática.